# Fast Discrete Fourier Spectra Attacks on Stream Ciphers

Guang Gong, Sondre Rønjom*, Tor Helleseth*, and Honggang Hu

Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1
*The Selmer Center
Department of Informatics
University of Bergen, PB 7803
N-5020 Bergen, Norway

**Abstract.** In this paper, we present some new results on the *selective discrete Fourier spectra attack*, introduced first as the recent *Rønjom-Helleseth attack* and the modifications due to Gong *et al.* The focal point of this paper is to fill some gaps in the theory of analysis in terms of discrete Fourier transform (DFT). We first analyze a special case of the selective DFT method which the previously introduced algorithm does not account for. We then proceed by introducing *fast selective DFT attacks*, which are closely related to the fast algebraic attacks in the literature. However, in contrast to the classical view that successful algebraic cryptanalysis of LFSR-based stream cipher depends on the degree of certain annihilators, we show that analysis in terms of the DFT spectral properties of the sequences generated by these functions is far more refined. It is shown that the fast selective DFT attack is more efficient than known methods for the case when the number of observed consecutive bits of a filtering sequence is less than the linear complexity of the sequence. Thus, by utilizing the natural representation imposed by the underlying LFSRs, we show in general that analysis in terms of DFT spectra is more efficient and has more flexibility than classical and fast algebraic attacks. Consequently, the selective DFT attack imposes a new criterion for the design of cryptographic strong Boolean functions, which is defined as the spectral immunity of a sequence or a Boolean function.

**Keywords:** Discrete Fourier Transform, linear feedback shift registers, filtering generators, fast algebraic attacks, spectral immunity.

## 1 Introduction

Linear feedback shift register (LFSR) sequences are widely used as basic functional blocks in key stream generators in stream cipher models due to their fast implementation in hardware as well as in software in some cases. Examples include filtering sequence generators, combinatorial sequence generators, clock-controlled sequence generators, and shrinking generators. The classical treatment can be found in [26]. In practice, E0 [7] and several of the submissions in eStream Project [11] are such examples. For an LFSR based stream cipher, the initial states of the LFSRs serve as a cryptographic key in each communication session. The goal of an attack is to recover the key from some known bits of the keystream. Consequently, the remaining bits of the keystream used in that session can be recovered and can be used in subsequent communication by only changing the known IV each time. There are many proposed attacks on LFSR based stream ciphers in the literature. However, in this paper, we will primarily restrict ourselves to the recently proposed algebraic attacks [1] [10], the fast algebraic attack, and its improvements or variants [2, 3, 12]. These attacks usually contain three steps: (a) pre-computation, (b) substitution for establishing a system of low-degree equations over $\mathbb{F}_2$ or $\mathbb{F}_{2^n}$ from known keystream bits, and (c) solving the system.

In 2003, Courtois [9] proposed the fast algebraic attack (FAA) on stream ciphers to accelerate the algebraic attack by identifying linear relations among the key stream bits. Compared with solving a system of equations directly by linearization and Gaussian elimination, the fast algebraic attack reduces the solving complexity by decreasing the total degree of the equation system, thus reducing the number of monomials in the system and the required number of keystream bits. The efficiency of the pre-computation and substitution in the fast algebraic attack is improved by Hawkes and Rose [12] for filtering sequence generators and Armknecht [2] for combinatorial sequence generators with or without memory, respectively. Along this line, Armknecht and Ars [3] introduced a variant of the FAA which reduced the number of required consecutive bits of the key stream, but leaves the number of unknowns unchanged. More recently, Rønjom and Helleseth [22] introduced the *linear subspace attack*, to recover the initial state of a filtering sequence generator. The attacker solves a linear system of $n$ equations in $n$ ($n$ typically 128 or 256) unknowns over $F_2$ after applying a function on $L$ bits of the keystream, where the total complexity of the attack is given by $O(L), L \leq D$, where $D$ is the number of monomials in the system. The attack is by far more efficient than the original algebraic attack and fast algebraic attack, but needs more keystream when compared to fast algebraic attack. Shortly after that, Rønjom, Gong, and Hellseth generalized the linear subspace attack by forming a system of linear equations over $\mathbb{F}_{2^n}$ instead [24], thus introducing more freedom in the attack which allows to cover some special cases where the original attack does not work directly.

In this paper we introduce *fast selective discrete Fourier transform (DFT) attacks*, which analogously to FAA-attacks seeks to reduce the number of unknowns in the resulting equation system. However, when interpreted in terms of the DFT of the keystream sequence we show that the resulting attack is more efficient than the standard FAA-algebraic attack. Furthermore, it works for the case when the known attacks fail, when the Boolean functions employed in filtering sequence generators are well-designed for meeting cryptographic requirements. This also imposes a new criterion for the design of key stream generators. As an analogue to the algebraic immunity of a Boolean function, the new criterion is referred to as the spectral immunity of a sequence or a Boolean function. In general speaking, algebraic attacks can be considered as attacks in the time domain, while (fast) selective DFT attacks are the attacks launched in the DFT frequency domain. The remarkable phenomenon is that with such an attack launched in the frequency domain, the number of unknowns in the system of linear equations is invariant. However, in the algebraic attacks or fast algebraic attacks, this number is changed at different time instances. We will provide the discussion of this phenomenon in details in Section 5.

This paper is organized as follows. Section 2 introduces basic definitions and results which will be used throughout this paper. In Section 3, we introduce the concept of selective DFT filters. Determining the DFT of a coordinate scaled sequence is equivalent to recovering the initial states of LFSRs based stream ciphers. In Section 4, we briefly reiterate the original algorithm from [24] for solving this problem given the DFT of the original sequence and $m$ consecutive bits of the coordinate scaled sequence for $m$ equal to the linear complexity of the sequence. Then we present an algorithm for solving this problem when $m$ is less than the linear complexity, referred to as a *fast selective DFT attack* in order to emphasize its relation to *fast algebraic attacks*. Section 5 shows the applications of these algorithms to filtering sequence generators, as well as the comparison with the known methods. A new criterion for the design of LFSR based stream ciphers for being resistant to these new attacks, called the spectral immunity of a sequence or a Boolean function, is introduced in Section 6, and some basic properties of spectral immunity are provided in the same section. Finally, Section 7 concludes this paper.

## 2 Preliminaries

### 2.1 Linear Feedback Shift Register (LFSR) Sequences

Let $t(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1x + 1$ be a polynomial over $\mathbb{F}_2$. A sequence $\mathbf{a} = \{a_t\}$ is called an *LFSR sequence* of degree $n$ if it satisfies the following recursive relation

$$a_{n+k} = \sum_{i=0}^{n-1} c_i a_{k+i}, k = 0, 1, \cdots . \quad (1)$$

$(a_0, \cdots, a_{n-1})$ is the *initial state* of the LFSR which generates $\mathbf{a}$. $t(x)$ is called a *characteristic polynomial of* $\mathbf{a}$, and the *reciprocal* of $t(x)$ is referred to as a *feedback polynomial of* $\mathbf{a}$. We also say that $\mathbf{a}$ is generated by $t(x)$. The sequence $\mathbf{a}$ is an $m$-sequences if $t(x)$ is primitive [14].

### 2.2 Minimal Polynomials, Linear Complexities, and the (Left) Shift Operator

The *minimal polynomial* of $\mathbf{a}$ is a polynomial with smallest degree which generates $\mathbf{a}$. Let $m(x)$ be the minimal polynomial of $\mathbf{a}$, then $m(x) \,|\, t(x)$. The *linear* *complexity (or linear span)* of $\mathbf{a}$ is the degree of $m(x)$, denoted by $l(\mathbf{a})$. In general, $m(x)$ can be found using the Berlekamp-Massey algorithm [4, 18] from any $2l(\mathbf{a})$ consecutive bits of $\mathbf{a}$ if the linear complexity is known. The *(Left cyclically) shift operator* $L$ is defined by $L\mathbf{a} = a_1, a_2, \cdots$, and $L^r\mathbf{a} = a_r, a_{r+1}, \cdots, r \geq 1$. If $\mathbf{b} = L^r\mathbf{a}$, then we say that they are *shift equivalent*, and $\mathbf{b}$ is a *shift* of $\mathbf{a}$. Otherwise, they are *shift distinct*. A sequence $\mathbf{a} = \{a_t\}$ is generated by $f(x)$ if and only if $f(L)\mathbf{a} = \mathbf{0}$, where $\mathbf{0}$ is the zero sequence.

### 2.3 The DFT and the Inverse DFT of Binary Sequences

Henceforth, we fix some notations for convenience.

- $N$: a positive integer;
- $n$: the smallest integer such that $N \,|\, 2^n - 1$;
- $\mathbb{F}_q$: a finite field with $q$ elements;
- $\mathbf{a} = \{a_t\}$: a binary sequence with period $N$ (it is not necessary that $N$ is the least period of $\mathbf{a}$);
- $l(\mathbf{a})$: the linear complexity of $\mathbf{a}$;
- $\alpha$: an element in $\mathbb{F}_{2^n}$ with order $N$;
- $n_s$: the smallest number $t$ such that $s \equiv s2^t \,(\mathrm{mod} N)$;
- $C_s$: the cyclotomic coset containing $s2^i$ modulo $N$, i.e., $C_s = \{s2^i \,(\mathrm{mod}\ N) \,|\, i = 0, 1, \cdots, n_s - 1\}$. The smallest number in $C_s$ is called the *coset leader* of $C_s$. Without loss of generality, we can assume that $s$ is the *coset leader* of $C_s$.
- $\overline{P}$: the set containing all the conjugates in $P$, i.e., $\overline{P} = \cup_{k \in P} C_k$, where $P$ is a subset of the set consisting of all coset leaders modulo $N$.

The *Discrete Fourier Transform (DFT)* of $\{a_t\}$ is defined by

$$A_k = \sum_{t=0}^{N-1} a_t \alpha^{-tk}, \ k = 0, 1, \ldots, N-1. \tag{2}$$

The *inverse DFT* is given by

$$a_t = \sum_{k=0}^{N-1} A_k \alpha^{kt}, \quad t = 0, 1, \ldots, N-1. \tag{3}$$

Note that the sequence $\{A_k\}$ is a sequence over $\mathbb{F}_{2^n}$, and it is called a *DFT spectral sequence* of $\mathbf{a}$ (with respect to $\alpha$) or DFT spectra for short. Let $A(x) = \sum_{k=0}^{N-1} A_k x^k$. Then $a_t = A(\alpha^t)$. Furthermore, $A(x)$ can be written into a more compact form as

$$A(x) = \sum_k Tr_1^{n_k}(A_k x^k) \tag{4}$$

and

$$a_t = \sum_k Tr_1^{n_k}(A_k \alpha^{tk}), t = 0, 1, ..., N-1 \tag{5}$$

where the $k$'s are coset leaders modulo $N$, $n_k \,|\, n$ is the size of $C_k$, and $Tr_1^{n_k}(x)$ is a trace function from $\mathbb{F}_{2^{n_k}}$ to $\mathbb{F}_2$. This is referred to as a *trace representation* of $\{a_t\}$. From now on, we may use $Tr(x)$ for all trace monomial terms in $A(x)$ for simplicity, where the exact meaning of $Tr(x)$, i.e., from which field to $\mathbb{F}_2$, depends on the size of the coset containing $k$. In this paper, we use (3), (4), and (5) interchangeably. Note that in the trace representation of $\mathbf{a}$, only the monomial trace terms which correspond to $A_k \neq 0$ are listed in the summation.

*Example 1.* Let $N = 15$, and $\alpha$ be a primitive element in $\mathbb{F}_{2^4}$ with $\alpha^4+\alpha+1 = 0$.

| Sequence | DFT Spectral Sequence | Trace Representation | Linear Complexity |
|---|---|---|---|
| $\mathbf{a}$ =000100110101111 | $\{A_k\}$=011010001000000 | $A(x) = Tr(x)$<br>$a_t = Tr(\alpha^t)$ | 4 |
| $\mathbf{b}$ =111011000101001 | $\{B_k\}$=$011\alpha\ 1\ 0\ \alpha^2\alpha^6 1\alpha^8\ 0\alpha^3\alpha^4\alpha^9\alpha^{12}$ | $B(x) = Tr(x + \alpha x^3 + \alpha^6 x^7)$<br>$b_t = B(\alpha^t)$ | 12 |

Let $\mathcal{N}_{\mathbf{a}} = \{k \,|\, A_k \neq 0, k \text{ is a coset leader mod } N\}$, $p_k(x)$ be the minimal polynomial of $\alpha^k$ over $\mathbb{F}_2$, and $p(x)$ be the minimal polynomial of $\mathbf{a}$. We use the notation $\mathcal{N}_{\mathbf{a}}^*$ to denote the set of nonzero coset leaders in $\mathcal{N}_{\mathbf{a}}$.

**Fact 1 ([6, 15])** *With the same notations as above, let $N = 2^n - 1$,*

1. *Conjugate property. $A_{k2^j} = A_k^{2^j}, j = 0, 1, \cdots, n_k - 1$.*
2. *Decomposition property. For each $k$ such that $A_k \neq 0$, let $\mathbf{a}_k = \{Tr(A_k\alpha^{tk})\}_{t\geq 0}$, then $\mathbf{a}_k$ is a sequence with period $N/\gcd(k, N)$, and the minimal polynomial of $\mathbf{a}_k$ is equal to the minimal polynomial of $\alpha^k$, i.e., $p_k(x) = \prod_{i=0}^{n_k-1}(x+\alpha^{2^i k})$, which is irreducible over $\mathbb{F}_2$ of degree $n_k$, where $n_k|n$.*
3. *If $\gcd(k, n) = 1$, then $\mathbf{a}_k$ is an m-sequence. Let $p(x)$ be the minimal polynomial of $\mathbf{a}$. Then $p(x) = \prod_{k,A_k\neq 0} p_k(x)$, and $\mathbf{a} = \sum_{k,A_k\neq 0} \mathbf{a}_k$.*
4. *The linear complexity of $\mathbf{a}$ is equal to the number of nonzero spectra of $\mathbf{a}$, i.e., $l(\mathbf{a}) = \sum_{k:A_k\neq 0} n_k$.*

### 2.4 The DFT of Coordinate Scaled Sequences

**Definition 1.** *If the sequence $\mathbf{b} = \{b_t\}$ is given by $b_t = A(\beta\alpha^t), t = 0, 1, \cdots, \beta \in \mathbb{F}_{2^n}$, then it is called a* coordinate scaled sequence *of $\mathbf{a}$.*

If the sequence $\mathbf{b} = \{b_t\}$ is a *coordinate scaled sequence* of $\mathbf{a}$, then the DFT of $\mathbf{b}$ is given by

$$B_k = \beta^k A_k, 0 \leq k < N \tag{6}$$

for some $\beta \in \mathbb{F}_{2^n}$, i.e., $B(x) = A(\beta x)$. Note that the coordinate scale operator does not change the minimal polynomial of $\mathbf{a}$, nor the linear complexity.

For the theory of LFSR sequences and their DFT spectra, the reader is referred to [14, 15].

### 2.5 Conversion between Solving the System of Linear Equations over $\mathbb{F}_{2^n}$ and $\mathbb{F}_2$

Let $P$ be a subset of the set of coset leaders modulo $N$, and $0 \notin P$. Now we consider the following system of equations with unknowns $Y_k$:

$$\sum_{k\in P} Tr_1^{n_k}(Y_k\gamma_{t,k}) = v_t,\ t = 0, 1, \cdots, |\overline{P}| - 1, \tag{7}$$

where $0 \neq \gamma_{t,k} \in \mathbb{F}_{2^{n_k}}$. Using the idea in [16], for any $k \in P$, $\{1, \alpha^k, \alpha^{2k}, ..., \alpha^{(n_k-1)k}\}$ is a basis of $\mathbb{F}_{2^{n_k}}$ over $\mathbb{F}_2$. Hence $Y_k$ can be written as the form of $x_{k,0} + x_{k,1}\alpha^k + ... + x_{k,n_k-1}\alpha^{(n_k-1)k}$, where $x_{k,i} \in \mathbb{F}_2$. Hence (7) can be converted into the following system of equations over $\mathbb{F}_2$ with $|\overline{P}|$ variables

$$\sum_{k\in P}\sum_{j=0}^{n_k-1} x_{k,j} Tr_1^{n_k}(\gamma_{t,k}\alpha^{jk}) = v_t,\ t = 0, 1, \cdots, |\overline{P}| - 1. \tag{8}$$

**Lemma 1.** *With the notations as above, if (7) has solutions for $Y_k, k \in P$, then $\{Y_k\}_{k\in P}$ can be computed by (8), the system of equations over $\mathbb{F}_2$, where $Y_k = \sum_{j=0}^{n_k-1} x_{k,j}\alpha^{jk}$.*

## 3 Selective DFT Filters

### 3.1 Single Sequence

The results presented below are fundamental for launching selective DFT or fast selective DFT attacks in the following sections.

**Lemma 2.** *Let $q(x) = \sum_{i=0}^{r} c_i x^i$ be a polynomial over $\mathbb{F}_2$ of degree $r$. Let $\mathbf{v} = q(L)\mathbf{a}$, and let $\{A_k\}$ and $\{V_k\}$ be the respective DFT spectra of $\mathbf{a}$ and $\mathbf{v}$. Then $V_k = A_k q(\alpha^k), 0 \leq k < N$.*

*Proof.* From the inverse of DFT given by (3), we have $a_t = \sum_{k=0}^{N-1} A_k\alpha^{tk}, t = 0, 1, \cdots$. Thus

$$v_t = \sum_{i=0}^{r} c_i a_{i+t} = \sum_{i=0}^{r} c_i \sum_{k=0}^{N-1} A_k\alpha^{(i+t)k} = \sum_{k=0}^{N-1} A_k\alpha^{tk}\sum_{i=0}^{r} c_i\alpha^{ik} = \sum_{k=0}^{N-1} A_k\alpha^{tk}q(\alpha^k).$$

□

According to Lemma 2 and Fact 1, we have the following cases for the DFT of $\mathbf{v}$.

(a) Two Extreme Cases:

- Case 1: $p(x) \mid q(x)$. Recall that $p(x)$ is the minimal polynomial of $\mathbf{a}$. Thus $q(\alpha^k) = 0$ for any $k \in \mathcal{N}_{\mathbf{a}}$. Hence $V_k = 0, 0 \leq k < N$ which means that $\mathbf{v}$ is the zero sequence.

- Case 2: $\gcd(p(x), q(x)) = 1$. Then $q(\alpha^k) \neq 0$ for any $k \in \mathcal{N}_{\mathbf{a}}$. Therefore $V_k = 0$ if and only if $A_k = 0$. In this case, $\mathbf{v}$ has the same minimal polynomial as $\mathbf{a}$.

(b) Selective Case: Let $c(x) = \gcd(p(x), q(x))$, $c(x) \neq 1$, and $c(x) \neq p(x)$. Then $c(x) = \prod_{k \in \mathcal{T}} p_k(x)$, where $\mathcal{T} \subset \mathcal{N}_{\mathbf{a}}$, and

$$V_k = q(\alpha^k) A_k \neq 0 \Longleftrightarrow q(\alpha^k) \neq 0 \Longleftrightarrow k \in \mathcal{N}_{\mathbf{a}} \setminus \mathcal{T}.$$

For the selective case, by applying $q(x)$ to $\mathbf{a}$, the nonzero DFT spectra of the resulting sequence is equal to a subset of the nonzero DFT spectra of $\mathbf{a}$. Thus the linear complexity of $\mathbf{v}$ is less than or equal to the linear complexity of $\mathbf{a}$. The functionality of $q(x)$ here is analog to filters used in communication systems for selecting frequency band for increasing or reducing bandwidth of transmitted signals. Thus $q(x)$ is referred to as a *selective DFT filter* of $\mathbf{a}$, and annihilates a subspace generated by some selected factors of the minimal polynomial of $\mathbf{a}$. We summarize the above discussion into the following theorem.

**Theorem 1.** *With the same notations as above, for* $t = 0, 1, \cdots$, *we have*

$$v_t = \begin{cases} 0, & p(x) \mid q(x); \\ \sum_{k \in \mathcal{N}_{\mathbf{a}}} Tr(q(\alpha^k) A_k \alpha^{tk}), & c(x) = 1; \\ \sum_{k \in \mathcal{N}_{\mathbf{a}} \setminus \mathcal{T}} Tr(q(\alpha^k) A_k \alpha^{tk}), & c(x) \neq 1, \ \textit{and} \ c(x) \neq p(x). \end{cases}$$

### 3.2 Product of Two Sequences

Let $\mathbf{s} = \{s_t\}$ and $\mathbf{b} = \{b_t\}$ be two sequences with period $N$, and $\mathbf{u} = \{u_t\}$ be a term by term product sequence defined by $u_t = s_t b_t, t = 0, 1, ..., N-1$. Let the DFT spectra of $\mathbf{b}$ and $\mathbf{u}$ be $\{B_k\}$ and $\{U_k\}$, respectively.

Suppose that $\mathcal{N}_{\mathbf{u}} \not\subset \mathcal{N}_{\mathbf{b}}$. Let $q(x) = \prod_{k \in \mathcal{N}_{\mathbf{u}} \setminus \mathcal{N}_{\mathbf{b}}} p_k(x)$, where $p_k(x)$ is the minimal polynomial of $\alpha^k$. Then the degree of $q(x)$ is $r = |\overline{\mathcal{N}}_{\mathbf{u}} \setminus \overline{\mathcal{N}}_{\mathbf{b}}|$. We denote $q(x)$ by $q(x) = \sum_{i=0}^{r} c_i x^i$. For $t = 0, 1, ..., l(\mathbf{b}) - 1$, put $f_t(x) = \sum_{i=0}^{r} c_i s_{i+t} x^i$. Applying $q(L)$ to $\mathbf{u}$, i.e., $q(L)\mathbf{u} = \mathbf{v}$, we have $v_t = \sum_{i=0}^{r} c_i u_{i+t}$ for $t = 0, 1, \cdots, l(\mathbf{b}) - 1$.

**Lemma 3.** *With the notations as above, we have*

$$\sum_{k \in \mathcal{N}_{\mathbf{b}}} Tr(B_k f_t(\alpha^k) \alpha^{tk}) = \sum_{k \in \mathcal{N}_{\mathbf{b}}} Tr(U_k q(\alpha^k) \alpha^{tk}), \ t = 0, 1, \cdots, l(\mathbf{b}) - 1.$$

*In particular, if* $q(x)$ *is the minimal polynomial of* $\mathbf{u}$, *then*

$$\sum_{k \in \mathcal{N}_{\mathbf{b}}} Tr(B_k f_t(\alpha^k) \alpha^{tk}) = 0, \ t = 0, 1, \cdots, l(\mathbf{b}) - 1.$$

*Proof.* For $t = 0, 1, \cdots, l(\mathbf{b}) - 1$, we have

$$\begin{aligned} v_t &= \sum_{i=0}^{r} c_i s_{i+t} b_{i+t} = \sum_{i=0}^{r} c_i s_{i+t} \sum_{k \in \mathcal{N}_{\mathbf{b}}} Tr(B_k \alpha^{(i+t)k}) \\ &= \sum_{k \in \mathcal{N}_{\mathbf{b}}} Tr(B_k f_t(\alpha^k) \alpha^{tk}). \end{aligned}$$

By Lemma 2, the result follows. □

## 4 Computing the DFT of Coordinate Scaled Sequences

Recall that $A(x) = \sum_k Tr_1^{n_k}(A_k x^k)$ and $a_t = A(\alpha^t)$. Let $\mathbf{s}$ be a coordinate scaled sequence of $\mathbf{a}$. In the following, we discuss how to compute the DFT of $\mathbf{s}$ given $\{A_k\}$ and $m$ consecutive bits of $\mathbf{s}$. From (6), we know that $S_k = \beta^k A_k$ for certain $\beta \in \mathbb{F}_{2^n}$, i.e., $s_t = A(\beta\alpha^t)$. Thus it is enough to find $\beta$. In the following, we first show how to solve this problem for $m = l(\mathbf{s})$ by rephrasing the algorithm from [22, 24] in terms of the DFT, and then present a new algorithm for the case of $m < l(\mathbf{s})$.

### 4.1 Selective DFT Method for $m = l(\mathbf{s})$

This case is already covered by the authors in [22, 24], but we repeat it here for the sake of completeness, and to cover a special case which has not been covered before. The sequence $\mathbf{s}$ has the same nonzero spectral index set as $\mathbf{a}$ since $\mathbf{s}$ is a coordinate scaled sequence of $\mathbf{a}$, i.e., $\mathcal{N}_{\mathbf{s}} = \mathcal{N}_{\mathbf{a}}$, so we use $\mathcal{N}_{\mathbf{s}}$ for both sequences.

**Algorithm 1** ALGORITHM FOR FINDING THE DFT OF COORDINATED SCALED SEQUENCES FOR $m = l(\mathbf{s})$

*Input:* $\{A_k\}$ *and* $m$ *consecutive bits of* $\mathbf{s}$.
*Output:* $\beta$, *the scalar factor in the DFT of* $\mathbf{s}$, *or* $S_k$, *a DFT spectral term of* $\mathbf{s}$.

**Pre-computation**

1. *Select* $k \in \mathcal{N}_{\mathbf{s}}$ *with* $|C_k| = n$, *compute* $p(x)$ *and* $p_k(x)$, *where* $p(x)$ *and* $p_k(x)$ *are the minimal polynomials of* $\mathbf{s}$ *and* $\alpha^k$, *respectively.*
2. *Set a selective DFT filter as* $q(x) = p(x)/p_k(x)$. *Hence* $q(x)$ *has degree* $r = l(\mathbf{s}) - n$, *and* $q(\alpha^i) = 0$ *for all* $i \in \mathcal{N}_{\mathbf{s}}, i \neq k$. *We write* $q(x) = \sum_{i=0}^{r} c_i x^i$.
3. *Compute* $q(\alpha^k)$.
4. *Compute the coefficient matrix of (8) in the case of* $P = \{k\}$: *i.e., compute* $h_{jt} = Tr(\alpha^{(j+t)k}), j = 0, 1, ..., n-1, t = 0, 1, ..., n-1$, *and set* $H = (h_{jt})_{n\times n}$.

**Procedure**

1. *Apply* $q(L)$ *to* $(s_0, s_1, \cdots, s_{l(\mathbf{s})-1})$, *i.e., we compute* $q(L)\mathbf{s} = \mathbf{v}$: $v_t = \sum_{i=0}^{r} c_i s_{i+t}, t = 0, 1, \cdots, n-1$.
2. *By Theorem 1, we have*

$$v_t = Tr(\theta\alpha^{tk}), t = 0, 1, \cdots, n-1, \text{ where } \theta = A_k q(\alpha^k)\beta^k. \tag{9}$$

*Solve the above system of* $n$ *linear equations with unknown* $\theta$ *in terms of solving*

$$H \cdot \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ \cdots \\ v_{n-1} \end{bmatrix}$$

*where* $\theta = \sum_{i=0}^{n-1} x_i \alpha^{ik}$ *by Lemma 1.*

3. *If* $\gcd(k, N) = 1$*, then* $\beta$ *can be computed by*

$$\beta = \theta^{k'}[A_k q(\alpha^k)]^{-k'}, \text{ where } k' = k^{-1} \pmod N.$$

*Return* $\beta$*. If* $\gcd(k, N) > 1$*, compute* $\beta^k = \theta[A_k q(\alpha^k)]^{-1}$ *and* $S_k = \beta^k A_k$*. Return* $S_k$*.*

The computation of $p(x)$, $p_k(x)$, $q(x)$ in Steps 1 and 2 in the pre-computation can be done by adopting the method in [2, 12]. One notable characteristic of Algorithm 1 is that the most cost of computation occurs in the pre-computation.

### 4.2 Selective DFT Method for $m < l(\mathbf{s})$

In this subsection, we introduce the fast selective DFT attack which is analogous to fast algebraic attack. From Lemma 3, we have the following algorithm to recover the scale factor $\beta$ when the number of known consecutive bits of $\mathbf{s}$ is less than $l(\mathbf{s})$.

**Algorithm 2** ALGORITHM FOR FINDING THE DFT OF COORDINATE SCALED SEQUENCES FOR $m < l(\mathbf{s})$

*Input:* $\{A_k\}$ *and* $m$ *consecutive bits of* $\mathbf{s}$*.*
*Output:* $\beta$*, the scalar factor in the DFT of* $\mathbf{s}$*, or* $\beta^k$ *for all* $k \in \mathcal{N}_{\mathbf{b}}$ *(*$\mathbf{b}$ *will be explained later.).*

**Pre-computation**

1. *Select* $\mathbf{b} = \{b_t\}$ *and* $\mathbf{u} = \{s_t b_t\}$ *which satisfy the following condition:*

$$|\overline{\mathcal{N}}_{\mathbf{b}} \cup \overline{\mathcal{N}}_{\mathbf{u}}| < l(\mathbf{s}). \qquad (10)$$

*(Note that here we only consider the case of* $B_0 = 0$ *and* $U_0 = 0$*. The other cases can be handled similarly. For simplicity, we omit them.)*

2. *Compute* $q(x) = \prod_{k \in \mathcal{N}_{\mathbf{u}} \setminus \mathcal{N}_{\mathbf{b}}} p_k(x)$*, and* $q(\alpha^k)$*, where* $k \in \mathcal{N}_{\mathbf{b}}$*. We denote* $q(x)$ *by* $q(x) = \sum_{i=0}^{r} c_i x^i$*. (Note that here we only consider the case of* $\mathcal{N}_{\mathbf{u}} \not\subset \mathcal{N}_{\mathbf{b}}$ *which is practical.)*

**Procedure**

1. *For* $t = 0, 1, \cdots, l(\mathbf{b}) - 1$*, using the known bits* $s_0, \cdots, s_{m-1}$*, compute* $f_t(\alpha^k)$ *for* $k \in \mathcal{N}_{\mathbf{b}}$*, where* $f_t(x) = \sum_{i=0}^{r} c_i s_{i+t} x^i$*.*
2. *Applying* $q(L)$ *to* $\mathbf{u}$*, by Lemma 3, we have*

$$\sum_{k \in \mathcal{N}_{\mathbf{b}}} Tr(\beta^k (B_k f_t(\alpha^k) + U_k q(\alpha^k))\alpha^{tk}) = 0, t = 0, 1, \cdots, l(\mathbf{b}) - 1. \qquad (11)$$

*if* $\mathcal{N}_{\mathbf{b}} \subset \mathcal{N}_{\mathbf{u}}$*, or*

$$\sum_{k \in \mathcal{N}_{\mathbf{b}}} Tr(\beta^k B_k f_t(\alpha^k)\alpha^{tk}) = 0, t = 0, 1, \cdots, l(\mathbf{b}) - 1, \qquad (12)$$

*if* $\mathcal{N}_{\mathbf{b}} \not\subset \mathcal{N}_{\mathbf{u}}$.
*Compute the coefficient matrix of (8) in the case of* $P = \mathcal{N}_{\mathbf{b}}$, *i.e.,* $Y_k = \beta^k$, $\gamma_{t,k} = (B_k f_t(\alpha^k) + U_k q(\alpha^k))\alpha^{tk}$ *for (11), and* $\gamma_{t,k} = B_k f_t(\alpha^k)\alpha^{tk}$ *for (12).*

3. *Solve the above system of* $l(\mathbf{b})$ *linear equations over* $\mathbb{F}_2$ *with* $l(\mathbf{b})$ *variables.*
4. *If there is a* $k \in \mathcal{N}_{\mathbf{b}}$ *with* $\gcd(k, N) = 1$, *then return* $\beta = (\beta^k)^{k'}$, *where* $k' = k^{-1} \pmod N$; *otherwise, return* $\{\beta^k \mid k \in \mathcal{N}_{\mathbf{b}}\}$.

In Algorithm 2, the number of required consecutive bits from $\mathbf{s}$ is at most $l(\mathbf{u}) + l(\mathbf{b})$.

The complexity in Step 1 of the pre-computation is contributed from the cost for finding the multiplier sequence $\mathbf{b}$ which satisfies (10). If there is no such a sequence, then this method won't work. Thus we exclude this case. This is a similar case as in the algebraic attack or the fast algebraic attack for finding a multiplier polynomial in which the cost for finding those multipliers are not counted [9].

In the following, we demonstrate this attack by one example.

*Example 2.* A generating polynomial of $\mathbb{F}_{2^5}$ is $x^5+x^3+1$. Let $\alpha \in \mathbb{F}_{2^5}$ satisfy $\alpha^5+\alpha^3+1 = 0$, and the trace representation $A(x)$ of $\mathbf{a}$ be $Tr(\alpha^{23}x^3+\alpha^{18}x^5+\alpha^{30}x^7+\alpha^{10}x^{11})$. The algebraic immunity of $A(x)$ is 3 which is maximum. Suppose that the key stream $\mathbf{s}$ we got is just the first 15 bits of $\mathbf{a} = \{A(\alpha^i)\}$: 000111111000110.

We select $\mathbf{b}$ and $\mathbf{u}$ whose trace representations are $Tr(\alpha^{28}x^3)$ and $Tr(\alpha^{30}x+\alpha^{16}x^{15})$ respectively. One can check that

$$Tr(\alpha^{23}x^3+\alpha^{18}x^5+\alpha^{30}x^7+\alpha^{10}x^{11})\cdot Tr(\alpha^{28}x^3) = Tr(\alpha^{30}x+\alpha^{16}x^{15}) \text{ for any } x \in \mathbb{F}_{2^5}$$

which means that $\mathbf{b}$ and $\mathbf{u}$ satisfy (10). We compute $q(x) = p_1(x)p_{15}(x) = x^{10} + x^8 + x^7 + x^5 + x^3 + x^2 + 1$, and $f_t(x) = \sum_{i=0}^{10} c_i s_{i+t} x^i$ for $t = 0, 1, 2, 3, 4$, where $\sum_{i=0}^{10} c_i x^i = q(x)$. Then we have

$$\begin{aligned}
f_0(x) &= x^8 + x^7 + x^5 + x^3, \\
f_1(x) &= x^7 + x^5 + x^3 + x^2, \\
f_2(x) &= x^{10} + x^5 + x^3 + x^2, \\
f_3(x) &= x^{10} + x^5 + x^3 + x^2 + 1, \\
f_4(x) &= x^8 + x^3 + x^2 + 1.
\end{aligned}$$

It follows that $f_0(\alpha^3) = \alpha^4$, $f_1(\alpha^3) = \alpha^{12}$, $f_2(\alpha^3) = \alpha^9$, $f_3(\alpha^3) = \alpha^{24}$, and $f_4(\alpha^3) = \alpha^6$. By (12), in order to get the scalar factor $\beta$, we need to solve the equations

$$\begin{cases}
Tr(B_3\beta^3\alpha^2) = 0, \\
Tr(B_3\beta^3\alpha^4) = 0, \\
Tr(B_3\beta^3\alpha^{15}) = 0, \\
Tr(B_3\beta^3\alpha^{18}) = 0.
\end{cases}$$

Let $B_3\beta^3 = x_0 + x_1\alpha + x_2\alpha^2 + x_3\alpha^3 + x_4\alpha^4$. Then we need to solve the equation

$$A \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

where

$$A = \begin{pmatrix} Tr(\alpha^2) & Tr(\alpha^3) & Tr(\alpha^4) & Tr(\alpha^5) & Tr(\alpha^6) \\ Tr(\alpha^4) & Tr(\alpha^5) & Tr(\alpha^6) & Tr(\alpha^7) & Tr(\alpha^8) \\ Tr(\alpha^{15}) & Tr(\alpha^{16}) & Tr(\alpha^{17}) & Tr(\alpha^{18}) & Tr(\alpha^{19}) \\ Tr(\alpha^{18}) & Tr(\alpha^{19}) & Tr(\alpha^{20}) & Tr(\alpha^{21}) & Tr(\alpha^{22}) \end{pmatrix} = \begin{pmatrix} 0\,0\,0\,1\,0 \\ 0\,1\,0\,1\,0 \\ 0\,0\,0\,1\,1 \\ 1\,1\,1\,1\,1 \end{pmatrix}.$$

The solutions are given by

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ or } \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

Hence $\beta = 0$ or 1, and $\beta = 1$ is the right answer indeed.

*Remark 1.* In Algorithm 2, we also can select $\mathbf{b}$ such that $\mathbf{sb} = \mathbf{0}$. In this case, we have a system of equations in $B_k\beta^k$ as

$$\sum_{k \in \mathcal{N}_{\mathbf{b}}} Tr(B_k\beta^k\alpha^{tk}) = 0$$

for $t$ such that $s_t = 1$. This is a similar case as the algebraic attack, and Algorithm 2 is a similar case as the fast algebraic attack. Those comparisons will be given in details in the next section.

## 5 Applications of the Selective DFT Attack to Filtering Sequence Generators

In this section, we show how to apply both algorithms presented in Section 4 for recovering an initial state of a filtering sequence generator and the comparison of the selective DFT attack with the known attacks. Let $\mathbf{w} = \{w_t\}$ be an $m$-sequence of period $N = 2^n - 1$ with trace representation $Tr(\beta x)$ and $w_t = Tr(\beta\alpha^t)$, $t = 0, 1, \cdots, \beta \in \mathbb{F}_{2^n}$, where $\alpha$ is a primitive element in $\mathbb{F}_{2^n}$. Let $0 \le d_0 < d_1 < \cdots < d_{r-1} < n$, where $r \le n$. A filtering sequence $\mathbf{s} = \{s_t\}$ with parameters $((\mathbf{w}, \beta), f, d_0, d_1, \cdots, d_{r-1})$ is defined by

$$s_t = f(w_{d_0+t}, w_{d_1+t}, \cdots, w_{d_{r-1}+t}),\ t = 0, 1, \cdots \qquad (13)$$

where $f(x_0, x_1, \cdots, x_{r-1})$ is a boolean function with $r$ variables. We denote $K = (k_0, \cdots, k_{n-1})$ as an initial state of $\mathbf{w}$, where $k_i \in \mathbb{F}_2$. Thus we have

$$s_t = f(L^t(k_0, k_1, \cdots, k_{n-1})) = f_t(k_0, \cdots, k_{n-1}), t = 0, 1, \cdots \tag{14}$$

where

$$f_t(x_0, \cdots, x_{n-1}) = f(L^t(x_0, \cdots, x_{n-1})).$$

Sometimes we also write $s_t = f(L^t(K))$ or $s_t = f_t(K)$ for short. If $K$ can be recovered from some known bits of $\{s_t\}$, then the remaining bits of $\{s_t\}$ can be reconstructed. Since $w_t = Tr(\beta\alpha^t), t = 0, 1, \cdots, n-1$, then recovering the initial state of the LFSR is equivalent to recover $\beta$. Let $H(k)$ denote the Hamming weight of $k$, and $\Gamma_k = \{\text{coset leaders } s \text{ modulo } N \mid H(s) \le k\}$. Then $|\Gamma_k| = \sum_{i=0}^{k} \begin{pmatrix} n \\ i \end{pmatrix}$. Furthermore, the DFT of $\{s_t\}$ has the following property: $S_k = 0$ for $H(k) > \deg(f)$. Thus

$$\overline{\mathcal{N}}_{\mathbf{s}} \subset \Gamma_{\deg(f)}. \tag{15}$$

Let $\mathbf{a}$ be the filtering sequence corresponding to $\beta = 1$ in the initial state of the LFSR. Then $\mathbf{s}$ is a shift of $\mathbf{a}$. Note that there are several ways for determining $\mathcal{N}_{\mathbf{a}}$ [22, 21]. So we assume that $\mathcal{N}_{\mathbf{s}}$ $(= \mathcal{N}_{\mathbf{a}})$ is known. Let $m$ represent the number of known consecutive bits of $\mathbf{s}$.

### 5.1 The Case of $m = l(\mathbf{s})$: Application of Algorithm 1

We assume that $\mathcal{N}_{\mathbf{s}}$ satisfies one of the following conditions.

(a) There is at least one $k \in \mathcal{N}_{\mathbf{s}}$ such that $\gcd(k, N) = 1$, or
(b) there are $t$ indexes $k_1, k_2, ..., k_t \in \mathcal{N}_{\mathbf{s}}$ such that $\sum_{i=1}^{t} a_i k_i (\text{mod } N)$ is coprime with $N$, where $a_i$ are nonzero integers.

By applying Algorithm 1 to $\mathbf{s}$, if the condition (a) is true, then we choose $q(x) = p(x)/p_k(x)$. So, Algorithm 1 returns $\beta$. Thus the initial state of the LFSR is recovered by $Tr(\beta\alpha^i), i = 0, 1, \cdots, n-1$. If the condition (a) is not true, but the condition (b) is true, we then select those $t$ indexes, and run Algorithm 1 $t$ times. Thus we obtain $\beta^{k_1}, \beta^{k_2}, ..., \beta^{k_t}$. Consequently $\beta = (\beta^{a_1 k_1} \beta^{a_2 k_2} ... \beta^{a_t k_t})^h$ where $h = (\sum_{i=1}^{t} a_i k_i)^{-1} \text{ (mod } N)$. So, the initial state of the LFSR is recovered.

Note that the pre-computation is only done once for the system. After this, the attacking complexity is the complexity of solving a system of linear equations over $\mathbb{F}_2$ in $n$ unknowns which is at most $O(n^w)$ provided that $l(\mathbf{s})$ consecutive bits of $\mathbf{s}$ is known for each communication session, where $w$ can be taken to be Strassen's reduction exponent $w = \log_2(7) \approx 2.807$. This is the most efficient attack currently known for the case that $l(\mathbf{s})$ consecutive bits of $\mathbf{s}$ are available to an attacker.

In the following, we interpret the results presented by Rønjom and Helleseth in 2006 [22] and Rønjom, Gong, and Helleseth in 2007 [24] under the language of the selective DFT method.

(i) In [22], $q(x) = p(x)/p_1(x)$. So, $q(L)$ removes all DFT spectra of $\mathbf{s}$ except for $S_1$ if it is not zero, i.e., $1 \in \mathcal{N}_\mathbf{s}$. Thus this attack will fail in some special cases when $S_1 = 0$, which occur with a very small probability.

(ii) In [24], $q(x) = p(x)/p_k(x)$. So, $q(L)$ removes all DFT spectra except for $S_k$ for some $k$ with $A_k \neq 0$ and $\gcd(k, N) = 1$. It will not work if there is no such $k$ in $\mathcal{N}_\mathbf{s}$.

(iii) The new result for a special case: if all spectral indexes in $\mathcal{N}_\mathbf{s}$ are not co-prime with $N$, but there are $t$ indexes $k_1, k_2, ..., k_t \in \mathcal{N}_\mathbf{s}$ such that $\sum_{i=1}^{t} a_i k_i (\bmod\ N)$ is coprime with $N$, where $a_i$ are nonzero integers. Thus the selective DFT attack works.

### 5.2 The Case of $m < l(\mathbf{s})$: Application of Algorithm 2

The selective DFT method of Algorithm 2 works if there are $\mathbf{b}$ and $\mathbf{u}$ which satisfy the condition (10). Here, we assume that there exists such sequences $\mathbf{b}$ and $\mathbf{u}$ in order to launch this attack. This is a similar case as in the algebraic attack or the fast algebraic attack on filtering sequence generators.

Using Algorithm 2, we obtain $\{\beta^k \,|\, k \in \mathcal{N}_\mathbf{b}\}$. Then we recover $\beta$. Consequently, $Tr(\beta\alpha^i), i = 0, 1, \cdots, n-1$, the required initial state of the LFSR or the key in the filtering sequence generator. It requires at most $l(\mathbf{b}) + l(\mathbf{u})$ consecutive bits of $\mathbf{s}$ to recover the initial state of the LFSR.

### 5.3 Comparison of the Selective DFT Attack with Fast Algebraic Attack and Its Variant

We first interpret the fast algebraic attack (FAA) introduced by Courtois in [9] and its variant by Armknecht and Ars in [3] by the language of the selective DFT method. Then we show the comparison of the new method with the FAA/variant. Let $M_r = \prod_{k \in \Gamma_r} p_k(x)$.

(i) In [9], $s_t g_t(K) = h_t(K)$, where $h = fg$ with $d = \deg(h)$, $e = \deg(g)$, and $d > e$, and $q(x) = M_d$, which results in $\mathbf{v} = 0$, i.e., $q(x)$ removes all the nonzero spectra of $\mathbf{u}$.

(ii) In [3], $q(x) = M_d/M_e$. In this case, $q(x)$ removes all possible nonzero spectra of $\mathbf{u}$ which do not belong to the set of possible nonzero spectra of $\mathbf{b}$ (note that $\overline{\mathcal{N}}_\mathbf{b} \subset \Gamma_e$ and $\overline{\mathcal{N}}_\mathbf{u} \subset \Gamma_d$ ).

(iii) For $\mathbf{u} = \mathbf{sb}$, in general, this results in a system of linear equations of unknowns $\beta^k$, where $k \in \mathcal{N}_\mathbf{b}$. By applying the selective DFT filter to both sides of the identity $\mathbf{u} = \mathbf{sb}$, the selective DFT filter remove the nonzero spectra of $\mathbf{u}$ which are not those of $\mathbf{b}$. Case (i) removes all nonzero spectra of $\mathbf{u}$. This is not necessary as we explained above. Case (ii) considered this problem. In Algorithm 2, we refine this using the DFT spectra of $\mathbf{u}$ and $\mathbf{b}$ instead of using $\Gamma_i$, where $i = d, e$.

(iv) The complexity of both FAA and the variant, not including pre-computation, is to solve a system of linear equations over $\mathbb{F}_2$ with $T_e$ variables. Algorithm 2 needs to solve a system of linear equations over $\mathbb{F}_{2^n}$ which can be converted into a system of linear equations over $\mathbb{F}_2$ with $l(\mathbf{b})$ variables. Note that the pre-computation only needs to be done once for the system. Thus the selective DFT is more efficient than the FAA and its variant if $l(\mathbf{b}) \ll T_e$.

The comparison of the selective DFT attack with the FAA and its variant is shown in Table 1 in which

- $m$ represents the number of the required consecutive bits of $\mathbf{s}$,
- $m^{'}$ represents the number of unknowns in a system of linear equations,
- $(x, y)$ represent solving a system of linear equations over $\mathbb{F}_x$ in $y$ unknowns, and
- $\delta = \begin{cases} 0 & \text{for FAA;} \\ 1 & \text{for FAA variants.} \end{cases}$

**Table 1.** Comparison between FAA/Variant and the Selective DFT Attack

| | $m$ | $m^{'}$ | $\deg(q)$ | $(x, y)$ |
|---|---|---|---|---|
| FAA/Variant | $T_e + (T_d - \delta T_e)$ | $T_e$ | $T_d - \delta T_e$ | $(\mathbb{F}_2, T_e)$ |
| Selective DFT<br>Attack | $l(\mathbf{b}) + \deg(q)$<br>The best case: $l(\mathbf{u})$ | $l(\mathbf{b})$ | $l(\mathbf{u}) - l(\mathbf{b}) \leq \deg(q) \leq l(\mathbf{u})$<br>where $l(\mathbf{u}) < T_d$ | $(\mathbb{F}_2, l(\mathbf{b}))$<br>The best case: $(\mathbb{F}_2, n)$ |

### 5.4 Difference between the Selective DFT Attack and FAA/Variant

(i) The FAA/variant works on a boolean function domain where the unknowns are monomials of $(k_0, k_1, \cdots, k_{n-1})$, an initial state of the LFSR which generates $\mathbf{s}$. So, one solves a system of linear equations over $\mathbb{F}_2$ with $T_e$ variables. On the other hand, the selective DFT attack works on a spectral domain, where the unknowns are $\beta$ and $Tr(\beta\alpha^i) = (k_0, k_1, \cdots, k_{n-1})$. So, one solves a system of linear equations over $\mathbb{F}_{2^n}$ which can be converted into a system of linear equations over $\mathbb{F}_2$ with $l(\mathbf{b})$ variables. Since $l(\mathbf{b})$ could be much smaller than $T_e$, the selective DFT attack could be more efficient than the former.

(ii) The coefficients of monomial terms in $b_t = g_t(x_0, x_1, \cdots, x_{n-1})$ with variables $x_0, x_1, \cdots, x_{n-1}$ are changed for each $t$, but the DFT spectra of $\{b_t\}$ are only changed by a scalar multiple of $\beta^k$, where $\beta$ corresponds to the desired initial state.

(iii) The number of nonzero coefficients of variables (linearized case) in $g_t(x_0, x_1, \cdots \cdots, x_{n-1})$ are dynamically changed for each $t$, which is bounded by $T_e$. Thus the number of unknowns in the system of linear equations cannot be reduced from $T_e$. However, the number of nonzero DFT spectral of $\{b_t\}$ remains a constant for all the shifts, which is the linear complexity of $\mathbf{b}$.

(iv) The phenomena (ii) and (iii) are not astonishing, since these are an analogue to the cosine function $\cos t$ which is hard to predict the values in real field. However, the Fourier transform of $\cos t$ has only two pulses (i.e., two values) which is a simplest case in spectral analysis.

(v) Another distinct difference between the FAA/variant and the selective DFT attack is that the filtering sequence $\mathbf{s}$ can be multiplied by any sequence with period $N$ ($N$ may not be the least period of $\mathbf{b}$) not just a filtering sequence. This opens a much wider window for this type of attacks, which will be addressed in the next subsection.

### 5.5 The Case of Boolean Functions with High Algebraic Immunity

The algebraic immunity of $f$ is defined as the smallest degree $\deg(g)$ such that $fg = 0$ or $(1 + f)g = 0$ [20], denoted as $AI(f)$. The FAA/variant works if there exist some functions $g$ and $h$ such that $fg = h$ and $e = \deg(g) < AI(f)$ (in the algebraic attack $h = 0$). From this result, the study for algebraic immunity of boolean functions is in fashion. If the algebraic immunity of $f$ is highest, i.e., $AI(f) = \deg(f)$, then for any $g$ such that $fg = h \neq 0$, if $e < AI(f)$, then $d = \deg(h) \geq AI(f)$. There are two cases as follows.

**Case 1.** $m \geq T_d$. Then FAA/variant works with the complexity of $O(T_e^w)$ operations in $\mathbb{F}_2$, where $w$ can be taken to be Strassen's reduction exponent $w = \log_2(7) \approx 2.807$. Since $l(\mathbf{s}) \leq T_{\deg(f)} \leq T_d$, Algorithm 1 is applicable. The latter is a much more efficient attack than the former when $T_e \gg n$.

**Case 2.** $m < T_d$. In this case FAA/variant is not applicable. However, from the selective DFT attack, what matters is the linear complexity $l(\mathbf{s})$. As long as $m \leq l(\mathbf{s})$, one can use the selective DFT attack of either Algorithm 1 for $m = l(\mathbf{s})$ or Algorithm 2 for $m < l(\mathbf{s})$ provided that there are such sequences $\mathbf{b}$ and $\mathbf{u}$ satisfying (10).

Thus, if the boolean function $f$ has high algebraic immunity, the selective DFT attack is either much more efficient than FAA/variant or applicable for those cases that FAA/variant fails.

The following is an important remark.

*Remark 2.* Under the DFT, the complexity of the sequence is measured by the number of its nonzero DFT spectra, which is the linear complexity of the sequence. This is the reason that the selective DFT attack could work for the case that $AI(f)$ is high but $m < l(\mathbf{s})$.

**A new criterion:** The above remark imposes a new criterion for the design of boolean functions with strong cryptographic properties. In other words, in addition to the existing criteria, a boolean function employed in a filtering or combinatorial generator should satisfy that there are no sequences $\mathbf{b}$ and $\mathbf{u}$ with low linear complexity such that (10) holds. This will be formally formalized in Section 6.

*Example 3.* The generating polynomial for $\mathbb{F}_{2^5}$ is $x^5 + x^3 + 1$. Let $\alpha \in \mathbb{F}_{2^5}$ satisfy $\alpha^5 + \alpha^3 + 1 = 0$, and $A(x) = Tr(\alpha^{27}x + \alpha^9 x^3 + \alpha^{14} x^7 + \alpha^7 x^{11})$. One can check that the algebraic immunity of $A(x)$ is 2. On the other hand, we have

$$A(x) \cdot Tr(\alpha^{29}x^5) = 0$$

for any $x \in \mathbb{F}_{2^5}$ which is the best case from the viewpoint of the selective DFT attacker.

## 6 Resiliency to the Selective DFT Attack and Spectral Immunity

From the results of Section 5.1, if $|\mathcal{N}_{\mathbf{s}}| = 1$, then Algorithm 1 is not applicable. This implies that in order to be resistant to the selective DFT attack of Algorithm 1, the minimal polynomial of an output sequence of an LFSR based key stream generator should be irreducible, which is another new criterion for the design of the LFSR based stream ciphers. Together with the discussions in Section 5.5, we have the following two cases which we consider as resistant to the selective DFT attacks. (We treat any vector in $\mathbb{F}_2^N$ as a binary sequence with period $N$.)

(a) Case 1. The minimal polynomial of $\mathbf{s}$ is irreducible, i.e., $|\mathcal{N}_{\mathbf{s}}| = 1$, then we say that it is *resistant to the selective DFT of Algorithm 1*;
(b) Case 2. The sequence $\mathbf{s}$ is said to be *resistant to the selective DFT of Algorithm 2* if $l(\mathbf{u}+\mathbf{b}) \geq l(\mathbf{s})$ for any sequence $\mathbf{b} \in \mathbb{F}_2^N$ with $l(\mathbf{b}) < l(\mathbf{s})$ and $\mathbf{u} \neq 0$, where $\mathbf{u}$ is the term-by-term product sequence of $\mathbf{s}$ and $\mathbf{b}$.

An example of sequence generators with a single nonzero spectral coset leader is the sequences generated by a stop-and-go clock control generator.

*Example 4.* Let $\mathbf{s}$ be a sequence generated by a stop-and-go clock control generator with the following parameters. Let $\mathbf{a} = \{a_t\}$ and $\mathbf{b} = \{b_t\}$ be two $m$-sequences with period $N = 2^n - 1$. Let $f(x)$ be the minimal polynomial of $\mathbf{a}$. Then

$$s_t = a_{w(t)}, \text{ where } w(t) = \sum_{j=0}^{t} b_j, t = 0, 1, \cdots$$

where the summation is the integer addition. Then the minimal polynomial of $\mathbf{s}$ is $f(x^N)$ which is irreducible over $\mathbb{F}_2$. Thus, the selective DFT attack of Algorithm 1 is not applicable.

In order to be resistant to the selective DFT attack from Algorithm 2, we introduce the concept of spectral immunity.

**Definition 2.** *Suppose that* $\mathbf{s}$ *is a binary sequence with period* $N$*. Let*

$$P_{\mathbf{s}} = \min_{\mathbf{b} \in \mathbb{F}_2^N} \{l(\mathbf{b}) \,|\, \mathbf{s} \cdot \mathbf{b} = \mathbf{0} \text{ or } (\mathbf{s}+\mathbf{1})\mathbf{b} = \mathbf{0}, \mathbf{b} \neq \mathbf{0}\}.$$

*Then* $P_{\mathbf{s}}$ *is referred to as the* spectral immunity *of* $\mathbf{s}$.

*Remark 3.* For a boolean function $f$, we may define its spectral immunity $P_f$ similarly using the one-to-one correspondence between boolean functions and sequences [15].

Let $P$ be the set consisting of all coset leaders modulo $N$. In the following, we present an algorithm to compute the spectral immunity of a given sequence. This algorithm is adopted from Algorithm 1 in [16].

**Algorithm 3** Algorithm for Finding the Spectral Immunity of a Binary Sequence

*Input: A nonzero binary sequence* $\mathbf{s}$ *with period* $N$.
*Output: The spectral immunity* $P_{\mathbf{s}}$ *of* $\mathbf{s}$.

```
1. Let $l = 1$.
2. Find all subsets $Q$ of $P$ such that $\sum_{k\in Q} n_k = l$. If there is no such subset,
   then go to Step 5.
3. For every $Q$, if there is a nontrivial solution $\{x_{k,i}\} \in \mathbb{F}_2^l$ such that
       $\sum_{k\in Q}\sum_{i=1}^{n_k} x_{k,i}Tr(\beta_{k,i}\alpha^{kt}) = 0$
   for any $t$ satisfying $s_t = 1$, then return $l$.
4. For every $Q$, if there is a nontrivial solution $\{x_{k,i}\} \in \mathbb{F}_2^l$ such that
       $\sum_{k\in Q}\sum_{i=1}^{n_k} x_{k,i}Tr(\beta_{k,i}\alpha^{kt}) = 0$
   for any $t$ satisfying $s_t = 0$, then return $l$.
5. Let $l = l + 1$, and go to Step 2.
```

We have the following upper bound for the spectral immunity of any binary sequence with period $N$.

**Theorem 2.** *For any nonzero sequence* $\mathbf{s}$ *with period* $N$*, let* $w(\mathbf{s})$ *denote the Hamming weight of* $\mathbf{s}$ *within one period. Let* $l$ *be the minimal integer such that* $l > \min(w(\mathbf{s}), w(\mathbf{s}+\mathbf{1}))$ *and there exists a subset* $Q$ *of* $P$ *satisfying* $\sum_{k\in Q} n_k = l$, *then* $P_{\mathbf{s}} \leq l$.

*Proof.* Without loss of generality, we can assume that $l > w(\mathbf{s})$. Then there is a nontrivial solution $\{x_{k,i}\} \in \mathbb{F}_2^l$ such that

$$\sum_{k\in Q}\sum_{i=1}^{n_k} x_{k,i}Tr(\beta_{k,i}\alpha^{kt}) = 0$$

for any $t$ satisfying $s_t = 1$ because the number of variables is bigger than the number of equations. Let $\mathbf{u} = \{u_t\}$ defined by $u_t = \sum_{k\in Q}\sum_{i=1}^{n_k} x_{k,i}Tr(\beta_{k,i}\alpha^{kt})$. Then $\mathbf{u}$ is nonzero, and $\mathbf{s}\cdot\mathbf{u} = 0$. So we have $P_{\mathbf{s}} \leq l$. □

## 7 Conclusions

Using the selective DFT method, we have presented a new algorithm analogous to the fast algebraic attack that works when the number of keystream bits $m$ is less than or equal to the linear complexity of the keystream. As applications of the algorithms, we show how to recover the initial states of LFSRs employed in filtering sequence generators when some consecutive bits of the key stream sequence are known. We have compared the methods with the fast algebraic attack by Courtois, and its variant by Armknecht and Ars for filtering sequence generators. The comparisons show that the selective DFT attack is more efficient and flexible when $m$ is less than or equal to the linear complexity of the sequence and where the employed boolean functions are well-designed for meeting cryptographic requirements such as having high algebraic immunity. Furthermore, we introduced a new concept named spectral immunity for cryptographic strong functions in order to be resist to the selective DFT attacks.

## Acknowledgement

The authors would like to thank the anonymous referees for their valuable and helpful comments.

## References

1. F. Armknecht and M. Krause, Algebraic attacks on stream combiners with memory, *Advances in Cryptology-CRYPTO 2003*, Lecture Notes in Computer Science, vol. 2729, pp. 162-176, Springer-Verlag, 2003.
2. F. Armknecht, Improving fast algebraic attacks, *Proceedings of Fast Software Encryption 2004*, Lecture Notes in Computer Science, vol. 3017, pp. 65-82, Springer-Verlag, 2004.
3. F. Armknecht and G. Ars, Introducing a new variant of fast algebraic attacks and minimizing their successive data complexity, Mycrypt 2005 (International Conference on Cryptology in Malaysia), Lecture Notes in Computer Science, vol. 3715, pp. 16-32, 2005, E. Dawson and S. Vaudenay (Eds.)
4. E. R. Berlekamp, *Algebraic coding theory*, New York, McGraw-Hill, 1968.
5. T. Beth and T. Piper, The stop-and-go generator, *Advances in Cryptology-Eurocrypt'1984*, Lecture Notes in Computer Science, pp. 88-92, Springer, 1984.
6. R.E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, 1983.
7. Bluetooth CIG, Specification of the Bluetooth system, Version 1.1, February 22, 2001. Available from www.bluetooth.com.
8. D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic programing, *Journal of Symbolic Computation*, vol. 9, pp. 251-280, 1990.
9. N. Courtois, Fast algebraic attacks on stream ciphers with linear feedback, *Advances in Cryptology-Crypto'2003*, Lecture Notes in Computer Science, vol. 2729, pp. 176-194, Springer-Verlag, 2003.
10. N. Courtois and W. Meier, Algebraic attacks on stream ciphers with linear feedback, *Advances in Cryptology-Eurocrypt'2003*, Lecture Notes in Computer Science, vol. 2656, pp. 345-359, Springer, 2003.

11. eSTREAM-*The ECRYPT Stream Cipher Project*, http://www.ecrypt.eu.org/stream/.
12. P. Hawkes and G. G. Rose, Rewriting variables: the complexity of fast algebraic attacks on stream ciphers, *Advances in Cryptology-Crypto'2004*, Lecture Notes in Computer Science, no. 3152, pp. 390-406, Springer-Verlag, 2004.
13. D. Gollmann, Pseudo random properties of cascade connections of clock controlled shift registers, *Advances in Cryptology-Eurocrypt'1984*, Lecture Notes in Computer Science, pp. 93-98, Springer, 1984.
14. S.W. Golomb, *Shift Register Sequences,* Holden-Day, Inc., San Francisco, 1967, revised edition, Aegean Park Press, Laguna Hills, CA, (1982).
15. S.W. Golomb and G. Gong, *Signal Design with Good Correlation: for Wireless Communications, Cryptography and Radar Applications*, Cambridge University Press, 2005.
16. G. Gong, Sequences, DFT and Resistance against Fast Algebraic Attacks, *Sequences and Their Applications (SETA)*, Lecture Notes in Computer Sciences, vol. 5203, S.W. Golomb, et al. (Eds.), Springer, 2008, pp. 197-218.
17. F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Publishing Company, 1977.
18. J. L. Massey, Shift-register synthesis and BCH decoding, *IEEE Trans. Information Theory*, Vol. 15, No. 1, pp. 122-127, January 1969.
19. J. L. Massey and S. Serconek, Linear complexity of periodic sequences: a general theory, *Advances in Cryptology-Crypto 96'*, Lecture Notes in Computer Science, vol. 1109, pp. 358-371, Springer-Verlag, 1996.
20. W. Meier, E. Pasalic, and C. Carlet, Algebraic attacks and decomposition of Boolean functions, *Advances in Cryptology, Eurocrypt 2004*, Lecture Notes in Computer Science, vol. 3027, pages 474-491, Springer-Verlag, 2004.
21. K.G. Paterson, Root Counting, the DFT and the Linear Complexity of Nonlinear Filtering, *Designs, Codes and Cryptography*, vol. 14, pp. 247-259, 1988.
22. S. Rønjom and T. Helleseth, A New Attack on the Filter Generator, *IEEE Transactions on Information Theory*, vol. 53, no. 5, pp. 17520-1758, 2007.
23. S. Rønjom and T. Helleseth, Attacking the Filter Generator over $GF(2^m)$, *Arithmetic of Finite Fields, First International Workshop, WAIFA 2007, Madrid, Spain, June 2007, Lecture Notes in Computer Science*, vol. 4547, pp. 264-275, 2007.
24. S. Rønjom, G. Gong, and T. Helleseth, On attacks on filtering generators using linear subspace structures, *Sequences, Subsequences, and Consequences*, *Lecture Notes in Computer Science*, S.W. Golomb *et al.* (Eds.), vol. 4893, pp. 204-217. Springer-Verlag, 2007.
25. S. Rønjom, G. Gong, and T. Helleseth, A survey of recent attacks on the filter generator, *the Proceedings of AAECC 2007*, pp. 7-17, 2007.
26. R.A. Rueppel, *Analysis and Design of Stream Ciphers*, Springer-Verlag, 1986.
27. R.A. Rueppel and O.J. Staffelbach, Products of linear recurring sequences with maximum complexity, *IEEE Transactions on Information Theory*, vol. 33, no. 1, pp. 124-131, January, 1987.